

Playback device and method for providing functionality based on event information retrieved from a playlist

The invention relates to a playback device for retrieving a data stream comprising video data comprising a java processor for processing an application, the java processor comprising an input for receiving an event information, to a Java processor for processing an application, the java processor comprising an input for receiving an event  
5 information and to a method for processing an java application.

Such a playback device is known from set top boxes complying with the MHP standard.

Such set topbox comprises a processor for processing an application, for instance a Java application.

10 The Java application provides a functionality to the set top box that is related to the data stream being played back by the set top box. For this the java application receives an event from the MHP video stream that indicates to the Java application that a certain position in the stream of video information is reached and that the associated functionality is to be provided by the Java application.

15 The event is stored in the video stream as a DSM-CC stream event.  
Storing the event in the stream has the disadvantage that the stream must be reprocessed if an event is to be changed.

It is an object of the invention to provide a method that allows changes to the events without extensive processing of the data stream and while still being able to provide  
20 event information at the appropriate position during playback of the video or audio data.

To achieve this objective the method is characterized in that the event information is retrieved from a playlist of the data stream.

By retrieving the event information from the play list that is associated with the data stream comprising video or audio data the event information is no longer retrieved  
25 from the data stream comprising the video or audio data. Since the event information is not comprised in the data stream, reprocessing of the data stream is not required and the data stream can remain unchanged when the event information is changed. In addition, by retrieving the event information from the play list a timing correlation between the playback of the video or audio information in the data stream and the event information can be

established. The playlist provides the playback device with information about when sections of the video or audio stream are to be played back. For instance a chapter mark indicating the start of a chapter can be used to activate functionality provided by a Java application that is related to this chapter. In this way the functionality associated to a chapter can be provided at the right moment, i.e. coordinated with the start of the playback of that chapter.

5 Changing event information requires the reprocessing of the playlist, which results in substantially less processing compared to the situation where the data stream must be reprocessed to change event information. In addition the playback device benefits from having the event information in the playlist because it no longer needs to demultiplex the event information from the data stream, reducing the required processing resources. An additional advantage is that the playback device is aware of the event information before the event arrives, because the playlist is retrieved before the events happen, and can thus schedule the launch of applications much better by anticipating the need to start the application and the anticipated processor work load at the moment of the start of the application and at the moment the event is reached during playback.

10 Hence the event information retrieved from the playlist allows the same functionality to be implemented as event information stored in the data stream itself, while avoiding the reprocessing of the data stream in order to change the event information. The object of the invention is consequently achieved.

20 An embodiment of the method is characterized in that the playlist comprises a mark with a presentation time and that the event information is information that the playback device reached the mark presentation time during playback.

The application needs to know when the functionality is to be provided.

25 The event information is retrieved from the playlist before the event is reached.

The application, now in the possession of the event information subsequently monitors the progress of the playback and provide the functionality when the playback has progressed to the point indicated in the playlist. The application then provides the functionality associated with the event.

30 Alternatively the event information can be provided to the application only at the moment the application must provide the functionality. The processor in the playback device retrieves event information from the playlist and only provides the event information to the application when the processor determines that the playback reached that point in the data stream corresponding to the event information in the playlist. Thus a regular application

can be used. The application does not need to monitor the progress of the playback of the data stream but relies on other processes running on the processor to monitor the play back of the data stream. Especially in the case of Java applications this is an advantage because the Java application does not need to be aware of lower level processes in the playback device  
5 and can be kept independent of the underlying hardware.

A playback device according to the invention is characterized in that the event information is received from a playlist of the data stream.

By retrieving the event information from the play list that is associated with the data stream comprising video or audio data the playback no longer retrieves the event information from  
10 the data stream comprising the video or audio data. Since the event information is no longer comprised in the data stream, reprocessing of the data stream is not required and the data stream can remain unchanged when the event information is changed. In addition, by retrieving the event information from the play list a timing correlation between the playback of the video or audio information in the data stream and the event information can still be  
15 established. The playlist provides the playback device with information about when sections of the video or audio stream are to be played back. For instance a chapter mark indicating the start of a chapter can be used to activate functionality provided by a Java application that is related to this chapter.

In this way the functionality associated to a chapter can be provided at the right moment, i.e.  
20 coordinated with the start of the playback of that chapter.

Changing event information requires the reprocessing of the playlist only, which results in substantially less processing compared to the situation where the data stream must be reprocessed to change event information. In addition the playback device benefits from having the event information in the playlist because it no longer needs to demultiplex the  
25 event information from the data stream, reducing the required processing resources. An additional advantage is that the playback device is aware of the event information before the event arrives, because the playlist is retrieved before the events happen, and can thus schedule the launch of applications much better by anticipating the need to start the application and the anticipated processor work load at the moment of the start of the  
30 application and at the moment the event is reached during playback.

Hence by retrieving the event information from the playlist the playback device is able to provide the same functionality as when the event information is stored in the data stream itself, while avoiding the reprocessing of the data stream in order to change the event information.

The object of the invention is consequently achieved.

An embodiment of the playback device is characterized in that the java processor comprises means for providing the event information to the application.

The application needs to know when the functionality is to be provided.

5 The event information is retrieved from the playlist before the event is reached.

The application, now in the possession of the event information subsequently monitors the progress of the playback and provide the functionality when the playback has progressed to the point indicated in the playlist. The application then provides the functionality associated  
10 with the event.

Alternatively the event information can be provided to the application only at the moment the application must provide the functionality. The processor in the playback device retrieves event information from the playlist and only provides the event information to the application when the processor determines that the playback reached that point in the  
15 data stream corresponding to the event information in the playlist. Thus a regular application can be used. The application does not need to monitor the progress of the playback of the data stream but relies on other processes running on the processor to monitor the play back of the data stream. Especially in the case of Java applications this is an advantage because the Java application does not need to be aware of lower level processes in the playback device  
20 and can be kept independent of the underlying hardware.

A further embodiment of the playback device is characterized in that the playlist comprises a mark with a presentation time and that the event information is information that the playback device reached the mark presentation time during playback. A mark can have a presentation time which is the time in the playback of the data stream  
25 when the presentation of a section of the data stream commences or stops. This an event. A functionality can be associated with this event. An application is used to provide this functionality.

A further embodiment of the playback device is characterized in that the mark is a chapter mark or a skip mark or a link mark.

30 Chapter marks, skip marks and link marks are already defined in the playlist.

It can be beneficial to provide functionality through a Java application to the user when a new chapter on the record carrier starts, or ends. For instance when an interactive record carrier complying with the DVD or blu-disk standard reaches a new chapter, the functionality may include displaying an interactive menu specially tailored to the video content of the chapter

reached. Similar functionality may be provided in association to the skip mark or the link mark.

A further embodiment of the playback device is characterized in that the mark is reserved for use by the application

- 5 Special marks may be inserted in the playlist. The special marks are not recognized by the playback device as regular playlist entries and thus current playback devices that do not comprise this invention can still correctly playback the information on the record carrier. Playback devices comprising the present invention recognize the special marks and provide the special marks to the Java application. All advantages of storing the event information in
- 10 marks in the playlist as discussed above are maintained when special marks are placed in and retrieved from the playlist while compatibility with the existing playback devices is maintained as well.

A further embodiment of the playback device is characterized in that the mark comprises further information for the application.

- 15 Application information may be appended to the mark. In that case the event information is derived from the mark itself while in addition the application information is provided to the application started by the event information. This allows more flexibility and more customization of the functionality provided by the application. Because current playback devices do not recognize the additional information the
- 20 additional information is ignore during playback and compatibility of a record carrier comprising additional information in the playlist for existing marks is achieved.

A Java processor according to the invention is characterized in that the event information is received from a playlist of a video stream.

- By retrieving the event information from the play list that is associated with
- 25 the data stream comprising video or audio data the playback no longer retrieves the event information from the data stream comprising the video or audio data. Since the event information is no longer comprised in the data stream, reprocessing of the data stream is not required and the data stream can remain unchanged when the event information is changed. In addition, by retrieving the event information from the play list a timing correlation
- 30 between the playback of the video or audio information in the data stream and the event information can still be established. The playlist provides the playback device with information about when sections of the video or audio stream are to be played back. For instance a chapter mark indicating the start of a chapter can be used to activate functionality provided by a Java application that is related to this chapter.

In this way the functionality associated to a chapter can be provided at the right moment, i.e. coordinated with the start of the playback of that chapter.

Changing event information requires the reprocessing of the playlist only, which results in substantially less processing compared to the situation where the data stream must be reprocessed to change event information. In addition the playback device benefits from having the event information in the playlist because it no longer needs to demultiplex the event information from the data stream, reducing the required processing resources. An additional advantage is that the playback device is aware of the event information before the event arrives, because the playlist is retrieved before the events happen, and can thus schedule the launch of applications much better by anticipating the need to start the application and the anticipated processor work load at the moment of the start of the application and at the moment the event is reached during playback.

Hence by retrieving the event information from the playlist the playback device is able to provide the same functionality as when the event information is stored in the data stream itself, while avoiding the reprocessing of the data stream in order to change the event information. The object of the invention is consequently achieved.

The invention will now be described based on figures.

Figure 1 shows a playback device comprising a java processor.

Figure 2 shows the application layers.

Figure 3 shows a flow chart of the method where the top level application layer monitors the progress of the playback of the data stream.

Figure 4 shows a flow chart of another embodiment of the method where the intermediate layer monitors the progress of the playback of the data stream.

Figure 1 shows a playback device comprising a java processor.

The playback device 2 is arranged for retrieving data, comprising a data stream, from the record carrier 1. The record carrier can be a DVD or a Blu-disk or any other record carrier comprising a data stream comprising video information and a playlist. The playback device comprises a basic engine 3 for retrieving the data from the record carrier 1. The basic engine 3 is connected to a processor 4 via a bidirectional interface. The processor can, via the bidirectional interface, instruct the basic engine to retrieve data from

locations on the record carrier 1 indicated by the processor 4. The processor 4 can thus instruct the basic engine 3 to retrieve a playlist from the record carrier 1 and to retrieve data comprising a data stream, or sections there of, from the record carrier 1. After the processor 4 received the playlist from the basic engine 3 the processor 4 retrieves event information from the playlist in a first section 7 of the processor 4 and provides monitors whether the playback of the record carrier reached the location of one of the events retrieved from the playlist. When the playback reaches the location of an event the first section of the processor provides the event information to a second section 6 of the processor that is used to run an application for providing a certain functionality when the location of a certain event is reached during playback. The application run by the second section 6 of the processor receives the event information and provides a functionality for instance in the form of video information to be displayed on a television set or monitor coupled to the playback device 2. In order to provide the functionality the second section 6 provides, in the example of video information, the video information to an output means 8 in the processor. The output means 8 provides the received video information obtained from the second section 6 to an output 9 of the playback device 2. The output 9 is connected to a television set or a monitor for viewing the video information.

The first section 7 comprises monitoring means to monitor the progress of the playback of the video information but can also comprise the decoding of the video information. The first section is in that case also coupled to the output means 8 in order to provide the video information to the output 9 of the playback device 2.

The output device can consequently, if provided with the video information of the functionality provided by the application and the video information obtained from decoding the video information in the data stream, output both at the same time, for instance by providing the video information from the data stream full screen and inserting the video information associated to the functionality provided by the application that received the event information into video information from the data stream. In case the functionality associated with the event provided by the application is a menu, the playback of the video information from the data stream can be halted until a choice from the menu is made. The menu can in that case be full screen and the video information from the data stream can be suppressed.

Figure 2 shows the application layers.

The hardware layer 20 is made independent of the top application layer 22 by an intermediate layer 21. Instructions from the top application layer, for instance a Java application are provided to the intermediate layer 21. The intermediate layer 21 translated the

instructions for the hardware layer 20, thus allowing the top application layer to be completely independent of the hardware layer 20.

As explained in figures 3 and 4 there are two alternative solutions for handling the event information.

5

- the top application layer 22 monitors the progress of the playback of the data stream
- the intermediate layer 21 monitors the progress of the playback of the data stream.

When the top application layer 22 monitors the progress of the playback of the data stream the top application layer 22 requests retrieval of the playlist from the record carrier. This request, given to the intermediate layer 21 is translated and the intermediate layer 21 requests the retrieval of the playlist by the hardware layer 20.

The hardware layer 20 retrieves the playlist from the recording medium and provides the playlist to the intermediate layer 21. The intermediate layer 21 then translates the playlist into the correct format for the top application layer 22. The top application layer 22 processes the playlist and retrieves the event information. Based on the event information the top application level 22 starts monitoring the progress of the playback by requesting playback progress status reports from the intermediate layer 21, which in turn request these playback progress status reports from the hardware layer 20. Once a playback progress status report is received, from the hardware layer 20 through the intermediate layer 21, indicating that the playback has progressed to the point in the data stream associated with the event derived from the event information, the top level application starts providing the functionality associated with the event.

When the intermediate layer 21 monitors the progress of the playback of the data stream the intermediate layer 21 requests retrieval of the playlist from the record carrier. The intermediate layer 21 requests the retrieval of the playlist by the hardware layer 20. The hardware layer 20 retrieves the playlist from the recording medium and provides the playlist to the intermediate layer 21. The intermediate layer 21 then extracts the event information from the playlist. Based on the event information the intermediate level 21 starts monitoring the progress of the playback by requesting playback progress status reports from the hardware layer 20. Once a playback progress status report is received indicating that the playback has progressed to the point in the data stream associated with the event derived from the event information, the intermediate level 21 provides the event information to the



top level application 22 which can then in turn start providing the functionality associated with the event.

Figure 3 shows a flow chart of the method where the top level application layer monitors the progress of the playback of the data stream.

- 5           In a first step 30 the top level application request the retrieval of the playlist. Once the playlist is retrieved the event information is extracted from the playlist in a second step 31. The event information is then provided to the top level application in a third step 32. Subsequently the top level application, in a fourth step 33, requests the processor, i.e. as explained an intermediate level application running on the processor, to monitor the progress of the playback of the data stream. This intermediate level application running on the processor monitors, in a fifth step 34, the progress of the playback of the data stream in a fifth step comprising a loop. The intermediate level application checks whether the playback has progressed to a certain point. If the playback has not reached the event location the intermediate application continues to monitor.
- 10           If the playback reached the event location a report is issued in the fifth step 34 to the top level application, the operation of the fourth step 33 continuing from this point and advancing to the sixth step 35 where the application starts providing the functionality associated with the event. Thus the event information provided in this case is the location of the event. The top level application is aware of the monitoring of the playback and is waiting, expecting a
- 15           trigger in the form of information about the status of the playback from another application that actually performs the monitoring.
- 20

Figure 4 shows a flow chart of another embodiment of the method where the intermediate layer monitors the progress of the playback of the data stream.

- In a first step 40 the top level application request the retrieval of the playlist.
- 25           Once the playlist is retrieved the event information is extracted from the playlist in a second step 41. The event information is then provided to the an intermediate level application in a third step 42. Subsequently the intermediate level application, running on the processor starts monitoring the progress of the playback of the data stream. The monitoring of the progress of the playback of the data stream in the fourth step 44 comprises a loop. The intermediate level application checks whether the playback has progressed to a certain point. If the playback has not reached the event location the intermediate application continues to monitor.
- 30           If the playback reached the event location a report comprising the event information retrieved from the playlits is issued in the fifth step 43 to the top level application. The method then advances to the sixth step 45 where the application starts providing the functionality

associated with the event. Thus the event information provided in this case is the actual reaching of the event by the playback. The top level application is not aware of the monitoring of the playback but gets a trigger in the form of the event information from another application that actually performs the monitoring.

5

A possible syntax for implementing the invention is shown below.

Proposed New Syntax

Syntax	No. of bits	Mnemonic
JavaPlayListMark() {		
<b>Length</b>	32	uimsbf
<b>number_of_PlayList_marks</b>	16	uimsbf
for(i=0; i < <i>number_of_PlayList_marks</i> ; i++) {		
Reserved	8	bslbf
<b>mark_type</b>	8	uimsbf
<b>ref_to_PlayItem_id</b>	16	uimsbf
<b>mark_time_stamp</b>	32	uimsbf
<b>entry_ES_PID</b>	16	uimsbf
<b>Duration</b>	32	uimsbf
Data_Bytes	8*16	Uimsbf
}		
}		

In this example the Data\_Bytes allows 16 bytes of data, this number is an example, less is sufficient for most cases.

10

Value	Meaning	Note
0x00	Reserved for future use	
0x01	Chapter-mark	See section X.X.X of application images. The <i>duration</i> field shall be set to zero. The <i>entry_ES_PID</i> shall be set to 0xFFFF. Data_Bytes are not defined in this case
0x02	Skip point	See section X.X.X of application images. The <i>duration</i> field shall be set to zero. The <i>entry_ES_PID</i> shall be set to 0xFFFF. Data_Bytes are not defined in this case
0x03	Link point	A mark referenced by a navigation command such as <i>Link MK</i> . When the player encounters a Link point in the process of a User Operation such as <i>Chapter Skip</i> , the player simply ignores the mark. The duration field shall be set to zero. The <i>entry_ES_PID</i> shall be set to 0xFFFF. Data_Bytes are not defined in this case
0x04 - 0x2F	Java Marks	A mark used by a Java Application
0x30 – 0xFF	Reserved for future use	

In this example, mark values from 0x04 to 0x2F are defined as Java marks.

5

The table below show the current definitions of marks that can be used as event information in the playlist. It also shows the values that are reserved for future use and that consequently can be used for the present invention.

## Mark Tables from BD-ROM Draft Spec

Syntax	No. of bits	Mnemonic
PlayListMark() {		
<b>Length</b>	32	uimsbf
<b>number_of_PlayList_marks</b>	16	uimsbf
for(i=0; i < <i>number_of_PlayList_marks</i> ; i++) {		
Reserved	8	bslbf
<b>mark_type</b>	8	uimsbf
<b>ref_to_PlayItem_id</b>	16	uimsbf
<b>mark_time_stamp</b>	32	uimsbf
<b>entry_ES_PID</b>	16	uimsbf
<b>Duration</b>	32	uimsbf
}		
}		

Value	Meaning	Note
0x00	Reserved for future use	
0x01	Chapter-mark	See section X.X.X of application images. The <i>duration</i> field shall be set to zero. The <i>entry_ES_PID</i> shall be set to 0xFFFF.
0x02	Skip point	See section X.X.X of application images. The <i>duration</i> field shall be set to zero. The <i>entry_ES_PID</i> shall be set to 0xFFFF.
0x03	Link point	A mark referenced by a navigation command such as <i>Link MK</i> . When the player encounters a Link point in the process of a User Operation such as <i>Chapter Skip</i> , the player simply ignores the mark. The <i>duration</i> field shall be set to zero. The <i>entry_ES_PID</i> shall be set to 0xFFFF.
0x03 - 0xFF	Reserved for future use	